

## CLAIMS

What is claimed is:

- 1 1. In a computer system executing a software stream that removes a block of heap memory  
2 from a heap pile for use by a hardware device, a method comprising:  
3 returning the block of heap memory to the heap pile by the hardware device.
- 1 2. The method as defined in claim 1 wherein returning the block of heap memory to the heap  
2 pile by the hardware device further comprises placing, in a linked list heap memory management  
3 system, the block of heap memory at an end of the linked list by the hardware device.
- 1 3. The method as defined in claim 2 wherein placing the block of heap memory at the end of  
2 the linked list by the hardware device further comprises:  
3 writing a null to a next block field of the block of heap memory;  
4 writing a block number of the block of heap memory to a next block field of a last block of  
5 heap memory in the linked list;  
6 changing the contents of a bottom register to point to the block of heap memory; and  
7 thereby  
8 making the block of heap memory a last entry in the linked list.
- 1 4. The method as defined in claim 3 further comprising returning, by the software stream, a  
2 second block of heap memory by placing the second block of heap memory at a beginning of the  
3 of the linked list.

1 5. The method as defined in claim 4 wherein placing the second block of heap memory at the  
2 beginning of the linked list by the software stream further comprises:  
3 determining a block number of a primary block of heap memory resident at the beginning  
4 of the linked list;  
5 writing the block number of the primary block of heap memory to a next block field of the  
6 second block; and  
7 writing atomically a block number of the second block to a top register.

1 6. The method as defined in claim 5 wherein determining a block number of a primary block  
2 of heap memory resident at the beginning of the linked list further comprises reading the top  
3 register prior to the step of writing the block number of the second block.

1 7. The method as defined in claim 2 further comprising removing, by the software stream,  
2 heap memory from the linked list heap management system.

1 8. The method as defined in claim 7 wherein removing heap memory from the linked list heap  
2 management system further comprises taking a primary block of heap memory resident at a  
3 beginning of the of the linked list.

1 9. The method as defined in claim 8 wherein taking a primary block of heap memory resident  
2 at a beginning of the of the linked list further comprises:  
3 determining a block number of the primary block;  
4 reading a next block field of the primary block of memory; and

5 removing the primary block if the next block field of the primary block does not indicate a  
6 null.

1 10. The method as defined in claim 9 wherein determining a block number of the primary  
2 block further comprises reading a top register, wherein the top register identifies the beginning of  
3 the linked list.

1 11. The method as defined in claim 9 wherein removing the primary block if the next block  
2 field of the primary block does not indicate a null further comprises writing a block number of the  
3 next block field of the primary block to the top register.

1 12. A method of managing a heap memory comprising:  
2 maintaining unused blocks of heap memory as a linked list, and wherein the unused blocks  
3 of the linked list comprise a first block at a beginning of the linked list, a second block pointed to  
4 the first block, and a third block at an end of the linked list;

5 removing, by a software stream, the first block from the linked list, thus making the second  
6 block the beginning of the linked list; and

7 returning, by a hardware device, a return block to the linked list by placing the return block  
8 at an end of the linked list.

1 13. The method of managing a heap memory as defined in claim 12 wherein returning, by a  
2 hardware device, a return block to the linked list by placing the return block at an end of the linked  
3 list further comprises:

4 writing a null to a next block field of the return block;  
5 reading a bottom register, the bottom register identifying the third block;  
6 writing a block number of the return block to a next state field of the third block; and  
7 writing the block number of the return block to the bottom register.

1 14. The method of managing a heap memory as defined in claim 12 wherein removing, by a  
2 software stream, the first block from the linked list further comprises:

3 reading a top register, the top register identifying the first block;  
4 reading a next block field of the first block, the next block field of the first block  
5 identifying the second block; and  
6 writing a block number of the second block to the top register.

7 15. The method of managing a heap memory as defined in claim 14 wherein writing a block  
8 number of the second block to the top register further comprises atomically writing the block  
9 number of the second block to the top register.

1 16. The method of managing a heap memory as defined in claim 12 further comprising  
2 returning, by a software stream, a fourth block to the linked list by placing the fourth block at the  
3 beginning of the linked list, thus making the fourth block the beginning of the linked list.

1 17. The method of managing a heap of memory as defined in claim 16 wherein returning a  
2 fourth block to the linked list by placing the fourth block at the beginning of the linked list further  
3 comprises:

4 reading a top register, the top register identifying the beginning of the linked list;  
5 writing a block number of the block identified by the top register to a next state field of the  
6 fourth block; and  
7 writing a block number of the fourth block to the top register.

1 18. The method of managing a heap memory as defined in claim 17 wherein writing a block  
2 number of the fourth block to the top register further comprises atomically writing the block  
3 number of the fourth block to the top register.

19. A method of managing a heap memory in a computer system, the method comprising:  
allowing a software thread to add and remove blocks of heap memory from a linked list of  
free blocks of heap memory in a last-in/first-out (LIFO) fashion at a first end of the linked list; and  
allowing a hardware device to add blocks of heap memory to the linked list of free blocks  
of heap memory at a second end of the linked list.

1 20. The method of managing a heap memory in a computer system as defined in claim 19  
2 wherein allowing a software thread to remove blocks of heap memory in LIFO fashion further  
3 comprises:  
4 determining, by the software thread, a block number of a block of heap memory at the first  
5 end of the linked list; and  
6 removing the block of heap memory at the first end of the linked list.

1 21. The method of managing a heap memory in a computer system as defined in claim 20  
2 determining a block number of a block of heap memory at the first end of the linked list further  
3 comprises reading a beginning register that stores a block number of a block of heap memory at the  
4 first end of the linked list.

1 22. The method of managing a heap memory in a computer system as defined in claim 21  
2 wherein removing the block of heap memory at the first end of the linked list further comprises:  
3 reading a next block field of the block of heap memory at the first end of the linked list to  
4 identify a block number of a next block in the linked list; and  
5 writing the block number of the next block in the linked list to the beginning register.

1 23. The method of managing a heap memory in a computer system as defined in claim 20  
2 wherein allowing a software thread to add blocks of heap memory in LIFO fashion further  
3 comprises:  
4 determining, by the software thread, a block number of a block of heap memory at the first  
5 end of the linked list;  
6 writing the block number of the block of heap memory at the first end of the linked list to a  
7 next block field of a return block of heap memory; and  
8 making the return block of heap memory the first end of the linked list.

1 24. The method of managing a heap memory in a computer system as defined in claim 23  
2 wherein determining a block number of a block of heap memory at first end of the linked list

3 further comprises reading a beginning register that stores a block number of a block of heap  
4 memory at the first end of the linked list.

1 25. The method of managing a heap memory in a computer system as defined in claim 24  
2 wherein making the return block of heap memory the first end of the linked list further comprises  
3 writing a block number the return block of heap memory to the beginning register.

1 26. The method of managing a heap memory in a computer system as defined in claim 20  
2 wherein allowing a hardware device to add blocks of heap memory to the linked list of free blocks  
3 of heap memory at a second end of the linked list further comprises:

4 determining, by the hardware device, a block number of a block of heap memory the  
5 second end of the linked list;

6 writing, by the hardware device, a block number of a return block of heap memory to a  
7 next block field of the block of heap memory at the second end of the linked list; and

8 making the return block of heap memory the second end of the linked list.

1 27. The method of managing a heap memory in a computer system as defined in claim 26  
2 wherein determining a block number of a block of heap memory at the second end of the linked list  
3 further comprises reading an end register that stores a block number of the block of heap memory  
4 at the second end of the linked list.

1 28. The method of managing a heap memory in a computer system as defined in claim 27  
2 wherein making the return block of heap memory the second end of the linked list further  
3 comprises writing a block number the return block of heap memory to the end register.

1 29. A computer system comprising:  
2 a microprocessor executing a software stream;  
3 a main memory array, a portion of the main memory array allocated to be a heap memory,  
4 and unused portions of the heap memory part of a heap pile;  
5 a first bridge logic device coupling the microprocessor to the main memory array;  
6 a hardware device coupled to the heap memory through the first bridge logic device;  
7 a graphics card coupled to the first bridge logic device, and also coupled to a video display  
8 device;  
9 wherein the software stream executed on the microprocessor removes blocks of heap  
10 memory from the heap pile for use by the hardware device; and  
11 wherein the hardware device returns blocks of heap memory to the heap pile without  
12 intervention from the software stream executed by the microprocessor.

1 30. The computer system as defined in claim 29 wherein the heap memory further comprises:  
2 a plurality of blocks, each block having the same number of bytes;  
3 each block having a next block field; and  
4 wherein the heap pile is maintained as a linked list, each block's next block field pointing  
5 to a next block in the list.



1 31. The computer system as defined in claim 30 wherein the linked list of blocks further  
2 comprises:

3 a top end being the beginning of the linked list;

4 a bottom end being the bottom of the linked list; and

5 wherein the software stream removes blocks from the heap pile by removing blocks from  
6 the top end, and the hardware device returns blocks to the heap pile at the bottom end.

1 32. The computer system as defined in claim 31 further comprising the software stream returns  
2 blocks to the heap pile at the top end.

3 33. The computer system as defined in claim 29 wherein the hardware device is the graphics  
4 card.

5 34. The computer system as defined in claim 29 wherein the hardware device is a network  
6 interface card.

1 35. The computer system as defined in claim 29 wherein the hardware device is an audio card.

1 36. The computer system as defined in claim 29 wherein the hardware device is a mass storage  
2 device.

1 37. The computer system as defined in claim 36 wherein the mass storage device is a hard  
2 drive.

1 38. The computer system as defined in claim 37 wherein the mass storage device is compact  
2 disk storage device.

TO BE FORWARDED TO THE PATENT OFFICE